Attorney's Docket No. 042933/301045 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.:	09/750,144	Confirmation No.: 6775
Applicant(s):	Steve Lewontin	
Filed:	December 29, 2000	
Art Unit:	2176	
Examiner:	Nguen Ba, Paul H.	
Title:	COMPACT TREE REPRESENTATION OF MARKUP LANGUAGES	

Docket No.: 042933/301045
Customer No.: 00826

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" filed April 11, 2006.

1. ***Real Party in Interest.***

The real party in interest in this appeal is Nokia Corporation, the assignee of the above-referenced patent application.

2. ***Related Appeals and Interferences.***

There are no related appeals and/or interferences involving this application or its subject matter.

3. ***Status of Claims.***

The present appeal involves Claims 1-21, which are presently under a final rejection as set forth by the final Official Action mailed October 14, 2005. A pre-appeal request was

submitted on April 11, 2006, but the decision of the panel of Examiners found that Claims 1-3, 5, 7, 10-13, 15, 17, 20, and 21 stand rejected because one or more issues are ripe for appeal. The claims at issue are set forth in the attached Claims Appendix.

4. ***Status of Amendments.***

No amendments have been filed subsequent to the final Official Action of October 14, 2005.

5. ***Summary of Claimed Subject Matter.***

The present patent application is directed to a compact tree representation of a document written in a markup language (e.g., XML). More particularly, independent Claim 1 provides a method of representing a document written in a markup language on a mobile terminal, and independent Claim 10 provides a mobile phone including software components for carrying out such a method. As recited, the method includes providing a virtual node tree describing the structure of the data types in the document but not containing actual document data, where each of the nodes in the virtual node tree respectively correspond to one element of a specific data type in the document. The method also includes providing a data array for each one of the nodes in the virtual node tree, where the data array includes information identifying the relationship of the node to other nodes in the virtual node tree and a reference indicating the location of data corresponding to the node. Thus, the data corresponding to the nodes using the reference included in the data array can be obtained by a set of software components in the mobile terminal.

6. ***Grounds of Rejection to be Reviewed on Appeal.***

The final Official Action and Advisory Action continue to reject Claims 1-3, 5, 7, 10-13, 15, 17, 20 and 21 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2004/0049737 to Simon Hunt et al., in view of U.S. Patent No. 6,003,033 to Amano et al. The final Official Action and Advisory Action also continue to reject Claims 4, 6, 8, 9, 14, 16, 18 and 19 under 35 U.S.C. § 103(a) as being unpatentable over the Simon Hunt

publication in view of the Amano patent, and further in view of U.S. Patent Application Publication No. 2002/0143521 to Call.

7. ***Argument.***

As background, the present patent application is directed to a compact tree representation of a document written in a markup language (e.g., XML). More particularly, independent Claim 1 provides a method of representing a document written in a markup language on a mobile terminal, and independent Claim 10 provides a mobile phone including software components for carrying out such a method. As recited, the method includes providing a virtual node tree describing the structure of the data types in the document but not containing actual document data, where each of the nodes in the virtual node tree respectively correspond to one element of a specific data type in the document. The method also includes providing a data array for each one of the nodes in the virtual node tree, where the data array includes information identifying the relationship of the node to other nodes in the virtual node tree and a reference indicating the location of data corresponding to the node. Thus, the data corresponding to the nodes using the reference included in the data array can be obtained by a set of software components in the mobile terminal.

The final Official Action and Advisory Action allege that the Simon Hunt publication discloses all of the elements of independent Claims 1 and 10 including providing a virtual node tree and obtaining data corresponding to the nodes of the tree, but does not disclose a virtual node tree not containing actual document data. For that element, the final Official Action cites the Amano patent. The final Official Action and Advisory Action then allege it would be obvious to one skilled in the art to modify the teachings of the Simon Hunt publication with those of the Amano patent to include a virtual node tree that does not include actual document data. Applicant respectfully disagrees on all three accounts.

A. Simon Hunt Cited Disclosure is not Prior Art to the Claimed Invention

With respect to the Simon Hunt publication, Applicant notes that the disclosure relied upon by the final Official Action for disclosing elements of the claimed invention is not prior art

to the present application, and can therefore not be properly cited against the claimed invention. In this regard, the Simon Hunt publication has a filing date of October 25, 2002 and claims priority as a continuation-in-part of two utility patent applications both filed April 25, 2001, as well as a provisional patent application filed April 26, 2000. The present application, on the other hand, has a filing date of December 29, 2000. The present application therefore has an effective filing date before the filing dates of the continuation-in-part application that was published on October 25, 2002, and the two utility patent applications on April 25, 2001. Accordingly, the Simon Hunt publication is only prior art for the subject matter that was first disclosed by the Simon Hunt provisional application and not subject matter first added in any of the subsequent utility applications. By way of example, only the content of the Simon Hunt provisional application that is carried over into the Simon Hunt publication may be considered prior art. Subject matter that is newly added in the Simon Hunt publication or utility applications that was not disclosed by the Simon Hunt provisional application is not prior art relative to the present application.

In order to determine the relevance of the Simon Hunt publication to the claimed invention, Applicant's undersigned attorney obtained and reviewed a copy of the Simon Hunt provisional application from the USPTO's public PAIR Web portal, attached hereto in the Evidence Appendix. The final Official Action cited portions (i.e., paragraphs 0146-0151) of the Simon Hunt publication directed to a QDOM module generating a representation of a document object model (DOM) tree of a document into an array that includes the start and stop positions of the document text as corresponding to the claimed feature of providing a virtual node tree. As explained in response to the final Official Action, however, the Simon Hunt provisional application, however, is silent as to the QDOM or its technique for generating a representation of a DOM tree. The Simon Hunt provisional application does disclose normalizing a DOM tree, but the disclosed normalization does not realize a data array including information identifying the relationship of a node to other nodes in the virtual node tree and a reference indicating the location of data corresponding to the node, as in the claimed invention. Thus, at least those portions of the Simon Hunt publication that were relied upon by the Official Action are not prior art relative to the claimed invention.

In response to the foregoing, the Advisory Action asserted that QDOM is the term for application of a DOM tree that identifies each node in a document using a unique value. The Advisory Action continued by asserting that the Simon Hunt provisional application discloses a DOM tree being subjected to a normalization process, citing the first paragraph of page 3 of the Simon Hunt provisional. Applicant respectfully submits, however, that in attempting to support the passages of the Simon Hunt publication cited for disclosing aspects of the claimed invention (i.e., paragraphs 0146-0151) the Advisory Action has not identified any corresponding passages of the Simon Hunt provisional application that actually support the cited passages and have, instead, only identified passages of the Simon Hunt provisional application that are directed to other steps of the overall process. That is, although the final Official Action cites passages of the Simon Hunt publication directed to the QDOM and its operation, the Advisory Action has not cited any corresponding portion of the Simon Hunt provisional application directed to the QDOM and its operation, but has instead cited passages of the Simon Hunt provisional application directed to the normalizer of the Simon Hunt publication.

Applicant respectfully submits that, although the Simon Hunt provisional application discloses subjecting a DOM tree to a normalization process, such a process is separate and distinct from the QDOM cited by the Examiner as disclosing aspects of the claimed invention. More particularly, the Simon Hunt publication discloses in part:

> Referring back to FIG. 2, the user agent 110 translates the requested data content, if necessary, into a recognizable markup language for further processing. The markup language may be in the format of XML, WML, HTML, or any other markup language or technology (e.g., video, audio, image) that incorporates the features used by the present embodiments.
>
> The translated information is then organized into a logically structured format for further processing by the QDOM 116. The QDOM 116 efficiently constructs a nodal structure. The use of the QDOM 116 enables a standard structured interface to the retrieved content that can be utilized by all modules of the server browser 110. The QDOM 116 can effectively and efficiently store the information content in a standardized structure for use by the normalizer [for normalization], more described below.

Simon Hunt Publication, paragraphs 0079-0080; and see paragraph 0085. Similarly, the Simon Hunt provisional application discloses in part:

> .... [A] Content Manager, preferably within the Gateway Cluster, translates the information into a markup language for further processing. This language may be eXtensible Markup Language (XML), HTML, or any language that incorporates the features used by the invention. The translated information is organized in a logically structured format for processing by a DOM Layer. This structure is typically a Document Object Model (DOM) tree.
>
> A Normalization Layer within the Content manager selects nodes from the DOM tree and analyzes elements of the tree. This analysis is used to create a new, normalized tree that can be efficiently accessed by an appliance.

Simon Hunt Provisional, pages 2-3.

As shown above, the Simon Hunt publication discloses translating content, which is organized into a DOM tree by a QDOM, and then subjected to a normalizer. The Simon Hunt provisional similarly discloses translating content, which is organized into a DOM tree by a DOM Layer (notably not by a QDOM), and then subjected to a Normalization Layer. The QDOM of the Simon Hunt publication therefore most closely corresponds to the DOM Layer of the Simon Hunt provisional. And separately and distinctly from the QDOM/DOM Layer, the normalizer of the Simon Hunt publication most closely corresponds to the Normalization Layer of the Simon Hunt provisional. Thus, the passages of the Simon Hunt provisional describing the Normalization Layer do not support or correspond to passages of the Simon Hunt publication describing the QDOM and cited as disclosing elements of the claimed invention.

As indicated above, one could argue that the DOM Layer of the Simon Hunt provisional corresponds to the QDOM of the Simon Hunt publication. Even in this instance, however, the Simon Hunt provisional does not describe the DOM Layer in such a manner so as to support the functions of the QDOM described in the Simon Hunt publication and cited against the claimed invention. In this regard, other than explaining that the DOM Layer organizes translated information into a DOM tree, the Simon Hunt provisional does not disclose any of the functions of the QDOM in the Simon Hunt publication cited against the claimed invention. In fact, throughout the Simon Hunt provisional, the input to the Normalization Layer (from the DOM Layer) is described as being a World Wide Web Consortium (W3C) DOM tree. Simon Hunt Provisional, page 15 (explaining that "[t]he W3C DOM Tree is the primary input to the Normalization Layer"). As explain in the Simon Hunt publication, on the other hand, the QDOM "extends the World Wide Web Consortium (W3C) DOM interface definition to an

efficient model that provides high speed parsing, storage, and access while minimizing memory resource requirements." Simon Hunt Publication, paragraph 0016; and *see id.* at paragraph 0157 (explaining that "[a] number of preliminary tests have been taken to determine the time saved using the QDOM 116 as compared to the node-based interface of the standard W3C DOM").

Accordingly, although the Simon Hunt provisional supports organizing information into a DOM tree, the Simon Hunt provisional does not support the functions of the QDOM cited by the final Official Action as disclosing elements of the claimed invention. The Simon Hunt provisional application does appear to support portions of the Simon Hunt publication directed to a normalizer. But nowhere does the Simon Hunt provisional teach or suggest that the normalizer realizes a data array including information identifying the relationship of a node to other nodes in the virtual node tree and a reference indicating the location of data corresponding to the node, as in the claimed invention.

B. The Amano Patent does not Teach/Suggest Elements of the Claimed Invention

As indicated above, the final Official Action and Advisory Action allege that the Simon Hunt publication discloses all of the elements of independent Claims 1 and 10 including providing a virtual node tree and obtaining data corresponding to the nodes of the tree, but does not disclose a virtual node tree not containing actual document data. For that element, as well as additional disclosure for the claimed provision of a data array for a node including information identifying the relationship of the node with other nodes and a reference including the location of data corresponding to the node, the final Official Action cites the Amano patent. Applicant respectfully disagrees.

The final Official Action alleges that the Amano patent discloses the provision of a data array for a node including information identifying the relationship of the node with other nodes and a reference including the location of data corresponding to the node, as recited by the claimed invention. For support, the final Official Action cites FIG. 18, and column 10, lines 29-40 of the Amano patent. As disclosed by the Amano patent, a data structure representing a tree is provided in the form of a table, where the table may be created by means of a text editor. Amano Patent, column 3, lines 36-58. The passage before the cited passage of the Amano patent

discloses that generating the data structure from the table includes, for each node of the corresponding tree, assigning a memory area to a node and setting attribute data for that node in the respective memory area. For child nodes of the tree, the cited passage of the Amano patent discloses connecting the child's memory area (area assigned to the child) to the parent's memory area (area assigned to the parent) by a pointer. The pointer is then set in an array for storing pointer data to child nodes, the pointer data including the address of the child node (address of the area assigned to the child) or an index of another array including the respective address.

The Amano patent therefore discloses generating a data structure whereby parent nodes are connected to child nodes via pointers. The Amano patent does not teach or suggest that the data structure includes, for each node of a virtual node tree, a reference indicating the location of data corresponding to the node, as recited by the claimed invention. One could argue that the Amano patent discloses, for a parent node, a reference indicating the location of data corresponding to a child node. The Amano patent does not teach or suggest, however, that the data structure for the parent node includes a reference indicating the location of data corresponding to the respective parent node, similar to the claimed invention. In fact, the Amano patent explicitly discloses that the attributes of the parent node are set in the memory area assigned thereto, at least suggesting that the Amano patent discloses that the memory area for the parent node includes the data corresponding to the parent node, as opposed to a reference to the respective data.

C. No Motivation to Combine Simon Hunt and Amano

Moreover, Applicant notes that even if one could interpret (albeit incorrectly) the Simon Hunt publication and Amano patent to disclose elements of the claimed invention to also be prior art to the claimed invention as alleged in the final Official Action, one skilled in the art would not have been motivated to combine the teachings of the Simon Hunt publication and Amano patent to teach or suggest the claimed invention. The final Official Action and Advisory Action appear to be applying impermissible hindsight in finding motivation to combine the the Simon Hunt publication and Amano patent to disclose the claimed invention. *See In Re Dembiczak*, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999) (explaining that "[c]ombining prior art references without

evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure of a blueprint for piecing together the prior art to defeat patentability -- the essence of hindsight").

As stated in MPEP § 2143.01, "the mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination." MPEP § 2143.01 (citing *In re Mills*, 916 F.2d 680 (Fed. Cir. 1990)) (emphasis added). In addition, as has been held by the Board of Patent Appeals and Interferences, and noted in the MPEP, the mere fact that one skilled in the art could adapt the reference device to meet the terms of a claim is not by itself sufficient to support a finding of obviousness. The prior art or the general knowledge of one skilled in the art must also provide a motivation or reason for one skilled in the art, without the benefit of applicant's specification, to make the necessary modifications to the reference device. MPEP 2144.04(VI.)(C.) (*citing Ex parte Chicago Rawhide Mfg. Co.*, 223 USPQ 351, 353 (Bd. Pat. App. & Inter. 1984).

Applicant respectfully submits that merely asserting that modifying the Simon Hunt publication with the Amano patent provides for easily describing a tree, and generating a data structure that corresponds to the tree in memory with the resultant description, without explaining the motivation or suggestion for such a combination (without relying upon hindsight or otherwise being guided by the present application), does not by itself render obvious the claimed invention. The Amano patent provides a system and method for generating a data structure using a data tree through a table. This tabulation of a data tree and generation of a data structure from the table, however, are separate and distinct issues from the generation of the data tree itself from an original document. One skilled in the art would not have been motivated to combine these distinct concepts because the generation of a data tree is a different issue from interpreting a data tree. In this regard, the ease with which the Amano patent interprets a data tree does not by itself provide motivation for one skilled in the art to modify the Simon Hunt publication to not only generate a tree, but also interpret its generated data tree in a manner similar to the Amano patent including the skeleton of a tree. That is, without an allegation that the Simon Hunt publication already discloses interpreting a generated data tree, or an allegation that one skilled in the art would be motivated to so modify the Simon Hunt publication, it is irrelevant how easy the Amano patent describes a tree and generates a data structure

corresponding thereto. In the instant case, however, Applicant respectfully submits that one skilled in the art would have no motivation to modify the Simon Hunt publication to interpret its generated data tree in a manner similar to that in the Amano patent, including use of the skeleton of a tree (alleged to correspond to the virtual node tree without actual document data).

Applicant further respectfully submits that the Call publication does not cure the defects of the Simon Hunt publication and Amano patent, and that independent Claims 1 and 10, and by dependency Claims 2-9 and 11-21, are therefore patentably distinct from the Simon Hunt publication, Amano patent and Call publication, taken individually or in combination. Accordingly, Applicant also respectfully submits that the rejections of Claims 1-21 as being unpatentable over various combinations of the Simon Hunt publication, Amano patent and Call publication should be reversed.

8. ***Claims Appendix.***

The claims subject to this appeal are as follows:

1. (Previously Presented) A method of representing a document written in a markup language on a mobile terminal, the method comprising:

providing a virtual node tree describing the structure of the data types in the document but not containing actual document data, each one of the nodes in the virtual node tree respectively corresponding to one element of a specific data type in the document;

for each one of the nodes in the virtual node tree, providing a data array including information identifying the relationship of the node to other nodes in the virtual node tree and a reference indicating the location of data corresponding to the node; and

obtaining, by a set of software components in the mobile terminal, the data corresponding to the nodes using the reference included in the data array.

2. (Original) The method recited in claim 1, wherein the data in the document is stored in a document block in memory.

3. (Original) The method recited in claim 2, wherein the document is written in XML or a variation of XML.

4. (Original) The method recited in claim 1, wherein the data arrays further include a flags field.

5. (Original) The method recited in claim 4, wherein a flag in the flags field indicates whether or not the node is the last sibling in a list of siblings.

6. (Original) The method recited in claim 4, wherein a flag in the flags field identifies the type of the node data.

7. (Original) The method recited in claim 1, wherein the relationship of the nodes to the other nodes in the virtual node tree is indicated by a child index and a sibling index in the data array.

8. (Original) The method recited in claim 1, wherein the data arrays have a fixed length.

9. (Original) The method recited in claim 1, wherein the data arrays have a variable length.

10. (Previously Presented) A mobile phone comprising:

a set of software components;

a memory connected to the set of software components; and

a display,

wherein at least one of the set of software components carries out a method of representing a document written in a markup language and rendering the document on the display, said method comprising:

providing a virtual node tree describing the structure of the data types in the document but not containing actual document data, each one of the nodes in the virtual node tree respectively corresponding to one element of a specific data type in the document;

for each one of the nodes in the virtual node tree, providing a data array including information identifying the relationship of the node to other nodes in the virtual node tree and a reference to the location of the data corresponding to the node; and

obtaining the data corresponding to the nodes using the references included in the data array.

11. (Original) The mobile phone recited in claim 10, further comprising a browser or other software application adapted to receive said document and render said document on said display.

12. (Original) The mobile phone recited in claim 10, wherein the document is an XML document and the browser is an XML browser.

13. (Original) The mobile phone recited in claim 10, wherein the data in the document is stored in a document block in said memory.

14. (Original) The mobile phone recited in claim 10, wherein the data arrays further include a flags field.

15. (Original) The mobile phone recited in claim 14, wherein a flag in the flags field indicates whether or not the node is the last sibling in a list of siblings.

16. (Original) The mobile phone recited in claim 14, wherein a flag in the flags field identifies the type of the node data.

17. (Original) The mobile phone recited in claim 10, wherein the relationship of the nodes to the other nodes in the virtual node tree is indicated by a child index and a sibling index in the data array.

18. (Original) The mobile phone recited in claim 10, wherein the data arrays have a fixed length.

19. (Original) The mobile phone recited in claim 10, wherein the data arrays have a variable length.

20. (Previously Presented) The mobile phone recited in claim 10, wherein the data arrays are stored in the memory of the mobile phone.

21. (Previously Presented) The method recited in claim 1, wherein the data arrays are stored in memory on the mobile terminal.

9. ***Evidence Appendix.***

An Appendix attached hereto includes the following Evidence:

U.S. Provisional Patent Application No. 60/199,858 entitled: *Method and Apparatus for Appliance Communication*, filed April 26, 2000.

10. ***Related Proceedings Appendix.***

None.

CONCLUSION

For at least the foregoing reasons, Applicant respectfully requests that the rejections be reversed.

Respectfully submitted,

Andrew T. Spence
Registration No. 45,699

CUSTOMER NO. 00826
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111

Evidence Appendix

1. U.S. Provisional Patent Application No. 60/199,858 entitled: *Method and Apparatus for Appliance Communication*, filed April 26, 2000.

Please type a plus sign (+) inside this box → +

Docket Number: P-6754RRCV

PROVISIONAL APPLICATION FOR PATENT COVER SHEET (Small Entity)

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Matthew	Trapani	Deerfield, IL
Leon	Polonsky	Wilmette, IL

☐ Additional inventors are being named on page 2 attached hereto

TITLE OF THE INVENTION (280 characters max)

METHOD AND APPARATUS FOR APPLIANCE COMMUNICATION

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☐ Customer Number → Place Customer Number Bar Code Label here

OR

☒ Firm or Individual Name: William T. Rifkin

Address	PIPER MARBURY RUDNICK & WOLFE				
Address	P. O. Box 64807				
City	Chicago	State	IL	ZIP	60664-0807
Country		Telephone	(312) 368-4000	Fax	

ENCLOSED APPLICATION PARTS (check all that apply)

☒ Specification — Number of Pages 25
☒ Drawing(s) — Number of Sheets 3
☐ Small Entity Statement
☐ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)

☐ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 19-2264

FILING FEE AMOUNT ($): $75.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE [signature] Date April 26, 2000

TYPED or PRINTED NAME Arik Ranson

REGISTRATION NO. (if appropriate) 43,874

TELEPHONE (312) 368-4000

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231

PROVISIONAL PATENT APPLICATION

Title: Method and Apparatus for Appliance Communication

Inventors: Matthew Trapani, Leon Polonsky

BACKGROUND OF THE INVENTION

This invention relates generally to the field of communications and, more particularly, to a method and apparatus for allowing appliances to communicate with networks, including, but not limited to, Internet, Extranet, and Intranet networks. The term "appliance" refers to any device or equipment, including, but not limited to: pagers; mobile phones; personal digital assistants (PDA's), whether or not they are wireless; vending machines; point of sale systems; home appliances, such as white goods (refrigerators, washers, dryers, etc.); office equipment; manufacturing equipment; or business equipment.

Currently, in order for users of devices, such as PDA's, to communicate with a network, content has to be configured for the particular device by the content producer. Parameters such as the device's operating system, capacity, or ergonomics of the device mandate customization of any content that is delivered to or from the device. Because it is cumbersome to tailor information to so many types of devices, this required customization restricts the availability of information that may be acquired or sent. Sometimes, the information is sent to the device, but in a capacity that the appliance, user, or network cannot conveniently accommodate. Also, some sites (such as some portals) attempt to provide a custom view of the web to a user, but the bulk of the customization still must be performed by the user. In addition, the portals only provide a high level customization of web links, etc. Other appliances, due to these and other restrictions, are not employed to receive or send information at all. With connectivity being increasingly desirable in today's world, a way of allowing appliances to more effectively communicate with networks is needed.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for appliance communication. The method and apparatus of the present invention allow end-to-end network communication infrastructures for both wireless and wired appliances. Any appliance, or group of appliances, can be configured to communicate with the Internet or other networks. The method and apparatus may be implemented on a server side, on a client side, or allocated between them. The communications involved may include text, audio, video, or any other type of data. The invention may communicate information both to and from an appliance. This information may be used for e-commerce, mCommerce, interactive browsing, or other applications.

The apparatus comprises an architecture containing a cluster configuration. The configuration includes a Gateway Cluster, which connects to networks such as the Internet, Intranet, and Extranet, or other connected networks, whether wired or wireless, or, more specifically, to one or more user agents within the network. This Gateway Cluster serves as a conduit for the appliances and the networks. The Gateway Cluster can perform the process of organizing and formatting the information to be sent to or from an appliance. These processes may be performed on a client or server side. The Gateway Cluster interacts with the appliances via a wireless network using any type of wireless network protocol.

In one embodiment of the present invention shown in figs. 1, 2, 3, and 6, information is received by the Gateway Cluster, typically including a central Session Layer from either an appliance or one or more networks. Connectivity managers may be used to compress, decompress, encrypt, or de-encrypt incoming or outgoing signals for communication with appliances. The Gateway Cluster , typically via a user agent, interfaces as a client with the network or networks containing content to be downloaded. When the Gateway Cluster downloads content from a network, a Content Manager, preferably within the Gateway Cluster, translates the information into a markup language for further processing. This language may be eXtensible Markup Language (XML), HTML, or any language that incorporates the features used by the invention. The translated information is organized in a logically structured format for processing by a DOM Layer. This structure is typically a Document Object Model (DOM) tree.

A Normalization Layer within the Content Manager selects nodes from the DOM tree and analyzes elements of the tree. This analysis is used to create a new, normalized tree that can be efficiently accessed by an appliance. The analysis includes parsing the elements (nodes) of the DOM tree via a filtering function to determine their priority (or "weight") based upon factors including, but not limited to, the font attributes of the element, the positional hierarchy of the element (within a document, table, row, etc.), or the frequency of that element being previously accessed by the appliance (or the user of that appliance). If the Normalization Layer determines that the element is to be added to the new tree, it is then added in a position relative to its weight. The information may run through an XML style sheet language (XSL) converter as part of an Artificial Intelligence Layer (Presentation Layer). The Presentation Layer determines how the information is to be displayed and returns the formatted device specific content, including media content, to the requesting device.

The highest level objects of the new, normalized tree are compressed, encrypted, and sent to the appliance. The Gateway Cluster acts as a server for delivering information to the appliance. If the appliance is wireless, the Gateway Cluster typically, though not necessarily, will communicate with the wireless network through XML, and the wireless network will in turn send the encrypted and compressed XML information to the appliance. The appliance is equipped with an Appliance Server, interfaced with the appliance preferably via a serial port. The Appliance Server integrates the incoming content, device applications, and media applications, and presents the information, typically in XML, to a browser in the appliance. This Appliance Browser interfaces with the Appliance Server to present information to the user and to send information, such as an event, to the Appliance Server to route back to the Gateway Cluster server. The information received by the Appliance Browser is then displayed on an appliance display, or is conveyed to the user in another manner, such as through a speaker.

The user may interface with the appliance via the Appliance Browser to request further objects. Active content downloaded from a network, but not yet sent to the appliance, is stored on the cluster server for sending to the appliance at the user's request. If the user of a device institutes an event (such as clicking on a link displayed on the device, for example), the event is sent to the session layer for generating a node query for further gathering of the higher level or lower level objects. Through interfacing with the Appliance Browser, which in turn interfaces with the

Appliance Server and the Cluster Server, the user may employ object-based browsing of the new object tree to receive lower level objects.

XML or other documents may also originate from the appliance. For example, a user may scan a product bar code with an appliance, produce an XML document with the Appliance Browser, and send the document via the Appliance Server and Cluster Server to an e-commerce transaction server for processing.

Further description of the invention may be had with reference to the drawings and documents provided with this filing.

Overview

InfoSpeed Communications has developed a unique methodology that enables appliances (handheld devices and equipment) to interact with Internet, Extranet and Intranet networks in a seamless fashion. Examples of appliances include 2-way pagers, mobile phones, Palm™ and Microsoft CE™ devices as well as wireless devices. Examples of equipment includes vending machines, point of sale systems and home appliances as well as office and manufacturing equipment. Unlike existing software solutions that allow appliances to access customized network information for wireless applications, InfoSpeed developed a software architecture that enables appliances to interact with existing network information without having to modify it. Currently, existing wireless internet systems provide content to handheld devices by requiring the content to be customized to the ergonomics of the device. However, InfoSpeed provides an end-to-end wireless internet access infrastructure solution that employs object based browsing capabilities.

Partial List of Features

1. Automatic translations - Translates existing HTML, XML and JavaScript (as well as their derivatives) "content" into an easy to use interface that adapts to any appliance type, size (screen size, available bandwidth, etc.) and operating environment in real-time.
2. Object based browsing - Architecture that breaks down existing network content and distributes the most relevant informational objects to the appliance
3. Guaranteed content delivery - Ensures that the information exchange between the device and network occurs without a single point of failure.
4. Object based personalization - A knowledge based system that provides the most frequently used web site pages and objects to the appliance.
5. XML based wireless infrastructure - XML is the protocol used to exchange information between the appliance and the network which enables appliances to perform e-commerce transactions.
6. Multipath wireless interactivity - This allows multiple appliances to interact with ancillary devices and media such as bar code scanners, GPS systems, voice, audio, and video in real time. Additionally, one or more appliances can send and receive information to one or more appliances and network servers in real-time.
7. MicroXML Browser - A microbrowser that operates on appliances using the XML protocol in an object based browsing architecture.

Product Characteristics

1. Supports all types of wireless networks regardless of network protocols (CDMA, GSM, TDMA, CDPD, 3G, UMTS, 802.11, Bluetooth, etc)

2. Supports stored and real-time voice, streamed and real time video and any type of data.

3. Performs the automatic translation algorithms using an adaptive technique that allows for content delivery

4. The system scalability functions as part of the system architecture and allows for load balancing techniques based on CPU and subscriber usage. It also allows the connectivity and content managers to run on a single platform as well as multiple servers within the cluster. The system is capable of supporting large numbers of subscriber records.

5. Quality of Service (QoS) delivery via guaranteed object-based browsing allowing for guaranteed e-commerce transactions.

6. Provides wireless security features that allow for secure transactions using standard digital certificate technologies.

7. Object based browsing reduces the size of the content that needs to be sent to the handheld device and thus allows for faster delivery of content.

8. Real-time personalized customization allows this system to provide an enhanced usability interface for a particular type of handheld device.

Network Description

InfoSpeed developed an architecture that resides on network servers in a cluster configuration and on an appliance (see figure 1). The cluster server connects to the content and media stored on the Internet, Intranet and Extranet as well as the wireless network that provides the wireless connectivity to the appliances. The Wireless Gateway Cluster consists of Connectivity and Content Managers. The connectivity managers is the exchange between the wireless network and the content managers. It performs the session state management, compression, load balancing and high availability features and addresses Feature 3 (Guaranteed Delivery). The Content Managers interact with existing content and performs the primary functions unique to this architecture such as Feature 1,2 & 4 (Automatic translations, Object based browsing and Object based personalization). In addition, the architecture uses XML as the protocol to exchange information between the various systems (Feature 5). Furthermore, the content managers support instant messaging capabilities between desktop browsers and multiple devices.

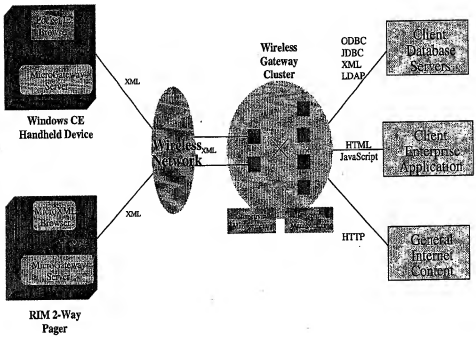

Figure 1: Network Diagram

Wireless Cluster Description

The Wireless Cluster appears to the network as a client interface, a PC that is browsing the web (see figure 2). However, when it downloads the content it creates an XML based Document Object Model (DOM) tree structure that organizes the content in a structured, simple format that can be interpreted and processed by the Artificial Intelligence Algorithms. These algorithms utilize the user's and device information to determine the most user friendly interface. It feeds the most relevant objects of the object tree in an orderly fashion to the appliance. Active content is content downloaded but not yet used by the user. It is stored on the server so that if the user requests an object, it can be accessed and processed. If a scripting language or media is within the content, the content interpreter will interact with the AI Presentation layer to perform the content operations and provide the relevant content. In addition, the appliance is validated by user/appliance ID which is stored within the Authentication database. This authenticates the appliance and validates that it is allowed to access specific content on a particular network. Thereafter, the XML objects are compressed & encrypted and sent to the wireless network. This is bi-directional in nature so if an XML request is initiated by the appliance, the server will process it and route it to the appropriate network element.

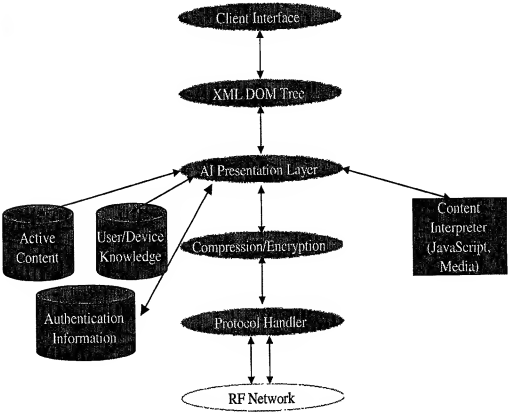

Figure 2: Wireless Cluster Content Manager Algorithm Diagram

AI Presentation Layer Description

The AI Presentation Layer (see figure 3) consists of four primary steps. It first identifies an object that is in the XML object tree to be processed, then it assigns a weight/priority to it based on its size, font and location relative to other objects. It also incorporates the frequency of use by an appliance in order to determine its importance to the appliance. The AI Layer records the number of times a particular user accesses a particular object within a web page. Thereafter, it determines how it fits within the object tree based on its hierarchy relative to other objects and is then linked in appropriately.

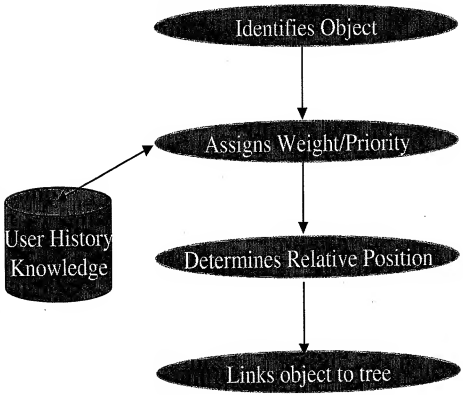

Figure 3: AI Presentation Layer Diagram

MicroGateway Description

The MicroGateway Server resides on the appliance as shown in figure 4. The function of the MicroGateway is to integrate incoming wireless content, device applications and media applications and allow it to be presented to a browser (Claim 6 - Multipath Wireless Interactivity). It interfaces with a browser using the HTTP/IP/XML protocols. It can allow GPS systems, bar code scanners, smart cards, voice, audio, video, bluetooth, infrared and other applications to be controlled by a browser performing object based browsing functions. The MicroGateway server first receives input from an application, performs the corresponding function associated with the application and then formats it XML so that it can interact with a browser.

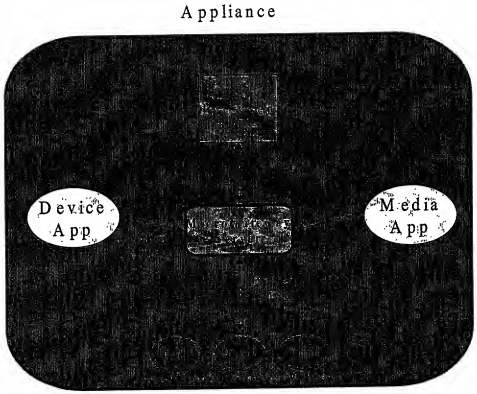

Figure 4: Appliance Diagram

MicroXML Browser Description

The MicroXML Browser resides on the appliance and interfaces with the MicroGateway Server. The function of the MicroXML Browser is to present information to the user and accept commands that the MicroGateway Server can accept and route back to the Wireless Cluster Servers. It is XML compliant and performs object based browsing functions (Claim 7). It first presents information to the device display. Thereafter, it receives input from the MicroGateway Server or device controls and performs the operations required to process the input.

Example

An example of an appliance that works with the InfoSpeed architecture is as follows. A user has a handheld device and wants to browse www.yahoo.com. The user enters in the URL on the MicroXML Browser which sends the request to the MicroGateway Server which compresses and sends the request to the Wireless Cluster Server. The server decompresses/de-encrypts the request, validates the appliance and then sends the operation to the most available Content Manager. The Content Manager then performs the request to the Internet and receives the www.yahoo.com content. Once received, it creates the object tree, determines the highest level objects and sends these compressed/encrypted objects to the user to choose the next level of objects. This can be a repeated operation. An example of an output from the algorithm is shown in Fig. 5.

Another example is when a user requests to purchase a product at a retail store using a handheld device. In this case, a user can first select their charge account for a particular store, bar code scan a product with the handheld and then originate an XML document that allows the user to purchase a product by sending the document to the e-commerce transaction servers.

Adaptive Browsing

We use statistics on events (onclick etc.) to move browsable objects around in the DOM tree. The AI Layer will keep statistics on the links that the user frequently hits in a user profile database. When the load type events for a link reach a threshold we move the object to the top of the DOM tree that is sent to the device. That way the user has access to the object at the top of the tree rather than having to walk the tree everytime or set a bookmark.

DOM Normalization Process (AI Layer)

1. Definitions.

1.1 **Node's Weight:** Positive or Negative integer that reflects the relative importance of the Node within the DOM Tree. The Higher the Weight the Higher that Node in hierarchy. The Root Node has the biggest weight. The Nodes Weight is the sum of **Node's Fontal Weight** and **Node's Positional Weight.**

1.2 **Node's Fontal Weight:** Importance of the Element Textual Data as represented by Font Attributes: Bold, Italic, All Caps, Background Color, Foreground Color, etc.

1.3 **Node's Positional Weight:** Reflects the positional hierarchy of the Node As defined by curtain HTML attributes: <LI> or <TABLE>. The Node's positional weight have a cumulative effect.
Example: Embedded <LI>: <LI> has a negative weight and having multiple layers of <UL>/<LI> sequence will have effect of embedded <LI> nodes will Be lighter than their parents.

1.4 **Filtering Process:** All textual elements of the original DOM go through The Filtering Function that might decide if the Node will have it's place on the AI Tree. Filter Function is also responsible for assigning the element All Caps Weight.
The elements that might filter out are:

1.4.1 Elements that contains only separator characters such as ".", ",", "|"
1.4.2 Any other filtering considerations.

1.5 **Node Positional Group Concept:** The group of elements that belong On the same "document line": such HTML tags as
, <P>, <TABLE>, <TR> and even <TD> will force the beginning of the New Element Positional Group.

1.6 **Node's Relative Position in Node Positional Group:** The sequential Number representing the Node sequence in the Positional Group:
Example:
<TD>
Pacific Rim: Asia Australia **China** HK Japan Korea
</TD>

"Pacific Rim" would be the first element, "China" – the fourth.

1.7 **Assignment of Node Fontal Weight**: The Normalization Process should Assign the Fontal Weight ONLY to the First Element in the Positional Group. In the Above example, only the Node "Pacific Rim" will gain weight, "China" will not.

1.8 **Node Table Locator**: The Normalization Process should keep a track of ALL Tables the Element is a part of: it's Row Position, Column Start and Column End. The Node Table Locator will play an important role in decision when to remove the Node from the AI Stack.

1.9 **Node New Element Identifier**: Some Web designers to represent something like:

MARCH SEARCH

Do:

<B>M</B>ARCH SEARCH

The Normalization Process should be able to recognize that and not make Node "ARCH SEARCH" the child of the Node "M".

1.10 **Scope Of Node Element:** Every DOM Element on the DOM tree has a Scope of Influence. While traversing the DOM Tree, every Node goes into Scope only once, and goes out of Scope only once.

2. Process.

2.1 The Normalization Process does it work by traversing the DOM Tree and creating A new normalized tree. The major part of the Process is maintaining the Weighted Node Stack. The First Element of that Stack is the "DOCUMENT" itself, having by default the highest possible weight (0xFFFFFFFF)

2.2 The Normalization Process takes the next Node off the DOM tree and puts it through **Filtering Function**. Filtering Function makes a decision whether that element belongs on the Normalized Tree or it should go into Scope of Influence. If the Node belongs to the Normalized Tree, the **Node's Weight** is computed by adding **Node's Fontal Weight** (according to 1.7) and **Node's Positional Weight.** Then Weighted Node finds its position on the **Weighted Node Stack** by finding the lightest element on the Stack with the Weight greater than his (**Node's Parent**). All Stack Nodes from that point on are deleted from the Stack. The New Weighted Node becomes a child Node for that Parent.

2.3 If the Node failed the Filtering Function, the decision is made whether that Node is added to **Influence Scope**. Such HTML Structural Elements as Frames, Tables, Table Rows and Table Data are added.

2.3.1 If the Stack Node was an element of the Table, and that table ends, Node Should be force-removed from the Stack.

2.3.2 If the Node Table Position reflects that it is NOT on the First Table Row and that Row goes, The Element should be Force-Removed.

2.4 When the Node goes out of Scope (TABLE is ended), the Normalization Process checks the **Weighted Node Stack** to remove all Nodes who belongs to **Expired Scope of Influence**. Example: If the Node on the **Weighted Node Stack** is part of the Table and Table Scope of Influence expired, that Node is removed from the **Weighted Node Stack.** However, if that Node belongs to more than One Scope of Influence (e.g. it is part of the Table inside the Table), ALL Scopes of Influences are checked against that Node and it is removed ONLY when they are all expired. In the example above, when the inner table ends, Node stays until it replaced by a heavier node or the Outer Table ends.

Infospeed Normalization and Presentation Layers

Purpose of this Document

This document specifies the Normalization layer of the PocketScape Media Manager which takes a HTML or XMLW3C DOM tree and utilizes a set of rules to group the DOM Nodes into hierarchical collections. The W3C DOM Tree is the primary input to the Normalization Layer. This will correspond to the W3C DOM1 Recommendation. The output of the Normalization Layer will be also be a W3C DOM Tree. The Presentation Layer is also described which transforms the W3c DOM tree into device specific formatting of the document.

Normalization Overview

The Infospeed Content Manager is required to serve HTML documents to devices in a way which provides the user with a more efficient/concise browsing mechanism. This is required for smaller display devices but it is also advantageous for use in fully capable browsers. The concept is that a User is able to browse only what he/she wants. This will cut down on bandwidth usage and response times. This browsing capability is referred to as Object Based Browsing.

Internet Portal Sites (myYahoo etc.) web sites attempt to provide a custom view of the web to a user. However the user must customize the site manually. Also the portals only provide a high level customisation of the Web links etc. In contrast, the Normalization Layer dynamically restructures HTML documents based on hieristic views of tags. It also dynamically restructures the HTML documents based on subscriber specific usage profiling (statistics).

The device is then provided the first layer of the tree when loading an HTML or XML document. The User than walks the DOM tree for specific information required. Higher level document objects are pushed to the top of the tree so the user can decide to walk down the branch or not.

Normalization Layer

The Normalization Layer applies rules to the markup in the document and groups certain markup into objects to be delivered to the device. The Normalization layer will produce the mapping between the Normalized DOM and the original W3C DOM.

Normalization Rules

1. Definitions.

1.1 **Node's Weight**: Positive or Negative integer that reflects the relative importance of he Node within the DOM Tree. The Higher the Weight the Higher that Node in hierarchy. The Root Node has the biggest weight.
The Nodes Weight is the sum of **Node's Fontal Weight** and **Node's Positional Weight**.

1.2 **Node's Fontal Weight:** Importance of the Element Textual Data as represented by Font Attributes: Bold, Italic, All Caps, Background Color, Foreground Color, etc.

1.3 **Node's Positional Weight:** Reflects the positional hierarchy of the Node As defined by curtain HTML attributes: <LI> or <TABLE>. The Node's positional weight have a cumulative effect.
Example: Embedded <LI>: <LI> has a negative weight and having multiple layers of <UL>/<LI> sequence will have effect of embedded <LI> nodes will Be lighter than their parents.

1.4 **Filtering Process**: All textual elements of the original DOM go through the Filtering Function that might decide if the Node will have its place on the Normalized DOM. Filter Function is also responsible for assigning the element All Caps weight.
The elements that might filter out are:

1.4.1 Elements that contains only separator characters such as ".", ",", "|"

1.4.2 Any other filtering considerations.

1.5 Node Positional Group Concept: The group of elements that belong On the same "document line": such HTML tags as
, <P>, <TABLE>, <TR> and even <TD> will force the beginning of the New Element Positional Group.

1.6 Node's Relative Position in Node Positional Group: The sequential Number representing the Node sequence in the Positional Group:

Example:
<TD>
Pacific Rim: Asia Australia **China** HK Japan Korea
</TD>

"Pacific Rim" would be the first element, "China" – the fourth.

1.7 **Assignment of Node Fontal Weight**: The Normalization Process should Assign the Fontal Weight ONLY to the First Element in the Positional Group.
In the above example, only the Node "Pacific Rim" will gain weight, "China" will not.

1.8 **Node Table Locator**: The Normalization Process should keep a track of ALL Tables the Element is a part of: it's Row Position, Column Start and Column End. The Node Table Locator will play an important role in decision when to remove the Node from the AI Stack.

1.9 **Node New Element Identifier**: Some Web designers to represent something like:

MARCH SEARCH

Do:

<B>M</B>ARCH SEARCH

The Normalization Process should be able to recognize that and not make Node "ARCH SEARCH" the child of the Node "M".

1.10 **Scope Of Node Element:** Every DOM Element on the DOM tree has a Scope of Influence. While traversing the DOM Tree, every Node goes into Scope only once, and goes out of Scope only once.

2. Process.

2.1 The Normalization Process does it work by traversing the DOM Tree and creating a new normalized tree. The major part of the Process is maintaining the Weighted Node Stack. The First Element of that Stack is the "DOCUMENT" itself, having by default the highest possible weight (0xFFFFFFFF)

2.2 The Normalization Process takes the next Node off the DOM tree and puts it through **Filtering Function**. Filtering Function makes a decision whether that element belongs on the Normalized Tree or it should go into Scope of Influence. If the Node belongs to the Normalized Tree, the **Node's Weight** is computed by adding **Node's Fontal Weight** (according to 1.7) and **Node's Positional Weight.** Then Weighted Node finds its position on the **Weighted Node Stack** by finding the lightest element on the Stack with the Weight greater than his **(Node's Parent)**. All Stack Nodes from that point on are deleted from the Stack. The New Weighted Node becomes a child Node for that Parent.

2.3 If the Node failed the Filtering Function, the decision is made whether that Node is added to **Influence Scope**. Such HTML Structural Elements as Frames, Tables, Table Rows and Table Data are added.

2.3.1 If the Stack Node was an element of the Table, and that table ends, Node Should be force-removed from the Stack.

2.3.2 If the Node Table Position reflects that it is NOT on the First Table Row and that Row goes, The Element should be Force-Removed

2.4 When the Node goes out of Scope (TABLE is ended), the Normalization Process checks the **Weighted Node Stack** to remove all Nodes who belongs to **Expired Scope of Influence**. Example: If the Node on the **Weighted Node Stack** is part of the Table and Table Scope of Influence expired, that Node is removed from the **Weighted Node Stack.** However, if that Node belongs to more than One Scope of Influence (e.g. it is part of the Table inside the Table), ALL Scopes of Influences are checked against that Node and it is removed ONLY when they are all expired. In the example above, when the inner table ends, Node stays until it replaced by a heavier node or the Outer Table ends.

URL Specific Normalization

The general normalization rules may not apply to all web sites. Therefore it may be possible to specify specific rules on a URL basis.

Statistic Adaptive Browsing/Normalization

The concept behind adaptive browsing is that most users hit certain links or objects in a document more than others. The user's statistics can be used to increase the priority of an object in the tree in order that the user be displayed that object first. This allows the page to be adapted to the user's tendancy/habits. This mechanism will aide in browsing efficiency for the user.

User Profile Adaptive Browsing/Normalization

A User can provide keywords for his profile which can be used to assign weights to the HTML document content. Content matching any keywords would be pushed to the top of the tree.

Presentation Layer

The Presentation Layer is responsible for taking a W3C DOM tree as input and producing a device specific media stream. This is performed by applying a style sheet or formatting rules to the DOM tree and outputting an XML document which will be streamed to the device.

Implementation

The preferred implementation of the Presentation Layer utilizes a XSLT (XSL Transformation) processor. This is preferred because of it's flexibility and it's standardized rule syntax. However it will also be possible to plug in a custom presentation layer processing. This custom presentation processing will be specific to a device or possibly URL.

References

http://www.alphaworks.ibm.com/formula/LotusXSL - IBM Commercial XSL processor
http://www.xml.com/pub/Guide/XSLT - list of XSLT Processor and APIs
http://www.w3.org/Style/XSL/ - XSL Specification

Figure 5

SAMPLE ALGORITHM OUTPUT

(Yahoo.com home page – output with proper indentation)

Yahoo!
Yahoo! Auctions
  Longaberger
  coins
  'N Sync
Free
  56K Internet Access
    advanced search
Shopping
Auctions
Yellow Pages
People Search
Maps
Travel
Classifieds
Personals
Games
Chat
Clubs
  Mail
Calendar
Messenger
Companion
My Yahoo!
News
Sports
Weather
TV
Stock Quotes
more...

Yahoo! Shopping

- Thousands of stores. Millions of products.

Departments

Stores

Products

Apparel

Bath/Beauty

Computers

Electronics

Flowers

Sports

Music

Video/DVD

Sports Authority

Ashford

Toys R Us

Nordstrom

'N Sync

Digital cameras

Pokemon

MP3 players

Arts & Humanities

Literature

Photography

Business & Economy

Companies

Finance

Jobs

Computers & Internet

Internet

WWW

Software

Games

Education

College and University

- K-12
- Entertainment
  - Cool Links
  - Movies
  - Humor
  - Music
- Government
  - Elections
  - Military
  - Law
  - Taxes
- Health
  - Medicine
  - Diseases
  - Drugs
  - Fitness
- News & Media
  - Full Coverage
  - Newspapers
  - TV
- Recreation & Sports
  - Sports
  - Travel
  - Autos
  - Outdoors
- Reference
  - Libraries
  - Dictionaries
  - Quotations
- Regional
  - Countries
  - Regions
  - US States
- Science
  - Animals

Astronomy
Engineering
Social Science
Archaeology
Economics
Languages
Society & Culture
People
Environment
Religion
In the News
Criticism of census form intensifies
Second giant iceberg breaks off Antarctica
NCAA women's final four
more...
Marketplace
Y! Tax Center
- file online with H&R Block
Free
56K Internet Access
Yahoo! Bill Pay
- free 3-month trial
more...
Inside Yahoo!
Y! Movies
- showtimes, reviews, info
Yahoo! Photos
- upload, share, and print pictures
Play free
Fantasy Baseball
Yahoo! GeoCities
- build your free home page
more...
World Yahoo!s
Europe

Denmark
France
Germany
Italy
Norway
Spain
Sweden
UK & Ireland
Pacific Rim
Asia
Australia & NZ
China
Chinese
HK
Japan
Korea
Singapore
Taiwan
Americas
Argentina
Brazil
Canada
Mexico
Spanish
Yahoo! Get Local
LA
NYC
SF Bay
Chicago
more...
Other
Autos
Careers
Digital
Entertainment

- Event Guide
- Greetings
- Health
- Invites
- Net Events
  - Message Boards
  - Movies
  - Music
  - Real Estate
  - Small Business
  - Y! Internet Life
  - Yahooligans!
- Yahoo! prefers
- How to Suggest a Site
- Company Info
- Privacy Policy
- Terms of Service
- Contributors
- Openings at Yahoo!

Figure 6

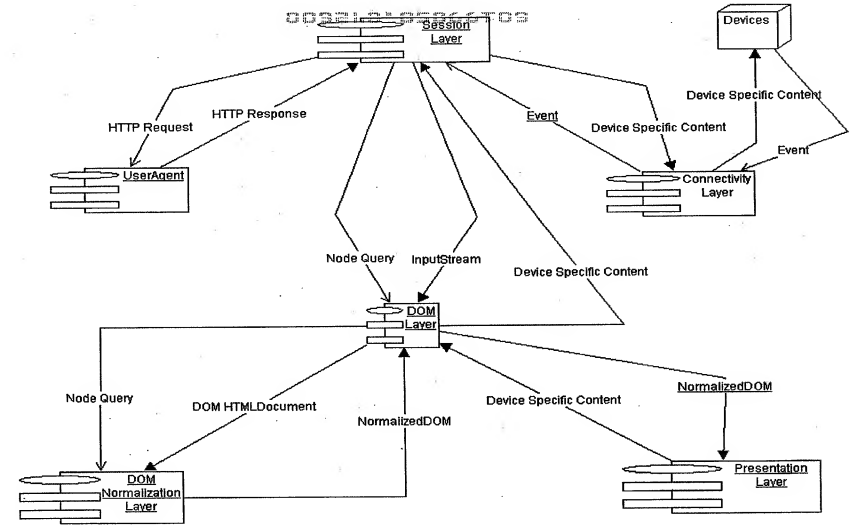
Session Layer
Devices
Device Specific Content
HTTP Request
HTTP Response
Event
Device Specific Content
Event
UserAgent
Connectivity Layer
Node Query
InputStream
Device Specific Content
DOM Layer
Node Query
DOM HTMLDocument
NormalizedDOM
Device Specific Content
NormalizedDOM
DOM Normalization Layer
Presentation Layer

Figure 7

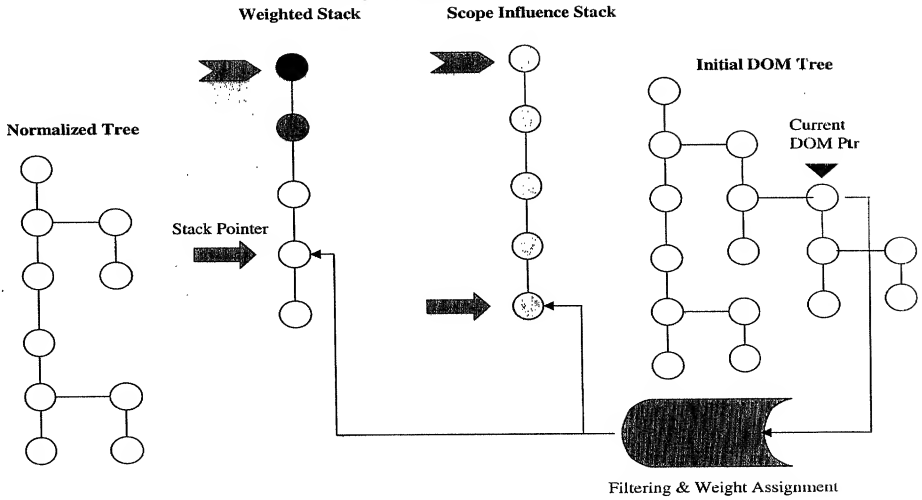
Weighted Stack
Scope Influence Stack
Initial DOM Tree
Normalized Tree
Current
DOM Ptr
Stack Pointer
Filtering & Weight Assignment

Figure 8

Flow Chart Decision Diagram